LCD DISPLAY BOARD · LCD280

2.8 inch SPI TFT display board

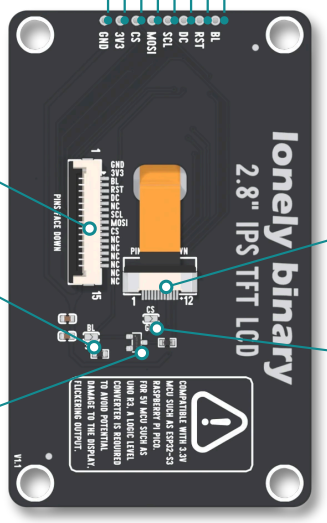
240 × 320 pixels, ST7789. One row of eight 2.54 mm holes, 3.3 V, and one library constant: LB_TFT_28.

3.3 V only CS, DC, RST, SCL and MOSI go straight to the controller, with no resistor and no level shifter, and no regulator sits behind the 3V3 hole. **No 5 V on any of them.** A 5 V board needs a level converter on those five lines and a 3.3 V rail for the 3V3 hole.

Eight holes, one set of names

Hole 1 is GND, at the left end seen from the front. The 15-way socket carries the same eight nets under its own numbers. A tinted hole can stay empty when its solder jumper is bridged.

	1	2	3	4	5	6	7	8
Front	GND	3V3	CS	MOSI	SCL	DC	RST	BL
15-way pin	1	2	9	8	7	5	4	3
ESP32-S3	GND	3V3	10	11	12	2	42	41
ESP32	GND	3V3	15	23	18	2	4	32



15-way socket U31
1.0 mm, pins face down.
The same eight nets as the row.

Jumper BL to 3V3
Open as shipped. Bridged: lamp always on. Leave BL unwired.

Backlight switch Q1
BSS138, in the lamp's low side.
10 kΩ holds it off until BL is HIGH.

Panel socket FPC1, 12 way
0.5 mm pitch, one per board.
The glass plugs here, not U31.

Jumper CS to GND
Open as shipped. Bridged: CS is held low. Leave CS unwired.

Dark ≠ dead A black screen with a good picture on it is a backlight nobody drove, not a dead panel. A 10 kΩ resistor holds the lamp off until **BL** is driven: tilt the screen under a bright lamp and the picture is there.

What a full redraw costs, at least

Pixels

76,800 240 × 320 RGB565, 2 bytes each

LB_TFT_28
ST7789 controller

One frame

153.6 kB over the wire for a full redraw

Every pixel, every time

Full redraw

≥ 30.7 ms on the wire at the library clock

40 MHz SPI clock

40 MHz	30.72
20 MHz	61.44

A full redraw cannot take less than 30.7 ms, which caps it near 32 full-screen frames a second. The real figure is slower: the controller's write time and your own drawing sit on top. Redraw only what changed and the cost falls with the area.

Arithmetic, not a measurement: pixels × 2 bytes × 8 bits ÷ the SPI clock the library's panel table gives this panel. No board was timed on this machine and neither host was tested, the ESP32-S3 or the classic ESP32. The second bar is the same redraw at half the clock, which long wires may force. Bars share one scale across all six boards: full length is 125 ms.

Your first picture, on the glass

```
/* First light: LCD280. Arduino IDE settings
   Board: ESP32S3 Dev Module, USB CDC On Boot: Enabled
   (or ESP32 Dev Module on a classic ESP32), Serial 115200 */
#include <LonelyBinaryDisplay.h>

LB_Display display(LB_TFT_28);  // the one line that changes

void setup() {
  Serial.begin(115200);
  delay(500);
  if (!display.begin()) {
    Serial.println("begin() failed: SPI bus did not start");
    return;
  }
  Serial.printf("%d x %d\n", display.width(), display.height());

  display.fillScreen(LB_BLACK);
  display.setTextColor(LB_WHITE);
  display.setTextSize(2);
  display.drawString("Hello", 8, 8);
  display.flush();            // always safe to call
}

void loop() {}
```

```
240 x 320
```

What Serial Monitor at 115200 should print: the size width() and height() report. The screen shows Hello in white on black.

To change the glass, change one token

BOARD	CONSTANT	PIXELS
0.96 inch	LB_TFT_096	80 × 160
1.8 inch	LB_TFT_18	128 × 160
2.0 inch	LB_TFT_20	240 × 320
2.4 inch	LB_TFT_24	240 × 320
2.8 inch	LB_TFT_28	240 × 320
3.5 inch	LB_TFT_35	320 × 480

Wiring

HOLE	NAME	BACK	15-WAY	ESP32-S3	ESP32
1	GND	8	1	GND	GND
2	3V3	7	2	3V3	3V3
3	CS	6	9	10	15
4	MOSI	5	8	11	23
5	SCL	4	7	12	18
6	DC	3	5	2	2
7	RST	2	4	42	4
8	BL	1	3	41	32

The same on every board of the family: the pins belong to the host and the glass to the constant. Both faces print the same names; **Back** counts the holes from BL, as the back reads. These are the library's defaults: `setWiring()` before `begin()` changes them.

The two solder jumpers

PAIR	BRIDGED	THEN
CS – GND	CS tied to GND: always selected	CS hole stays empty; no other device may share the bus
BL – 3V3	BL tied to 3V3: lamp always on	BL hole stays empty; no dimming

Both ship open. Do not drive a pad whose jumper is bridged: the pin would fight a rail. A meter on continuity across the pair tells you what your board has.

When it does not work

The screen is dark but a picture shows at an angle

The panel is drawn on and the lamp is off. BL is unwired, never driven, or set to zero; the 10 kΩ holds the lamp off until it is. Check BL driven HIGH. The other seven holes are fine.

begin() returns false

The SPI bus itself would not start, which is almost always a pin number the chip does not have (a C3 on the defaults for the other two). A wiring fault on DC, RST or CS cannot make it return false: the panel is written to and never read, so begin() returns true with a blank screen.

It flickers, or the picture tears halfway down

The signal arrives but not cleanly: wires too long, the clock too fast, or a 5 V board on a 3.3 V input. Shorten the wires, call setSpiHz(20000000) before begin(), and check the logic level of the host.

It is plain white and has never shown anything

Many panels power up white: the controller is alive and nothing has reached it. Check DC, then RST, then the 15-way cable if you use it, contacts down. Power, ground and the lamp are proven.

Red comes out blue, or it is a photo negative

Colour order and inversion are one call each: setColorOrder() and setInverted(), before or after begin(). Both at once turns red into yellow. The right values belong in the library's table: report them.

Nothing works on an ESP32-C3

The library's default pins are for the ESP32-S3 and the classic ESP32; GPIO 23 and 32 do not exist on a C3. Copy LB_WIRING, change the pins to ones the C3 has, and call setWiring() before begin().

Specifications

Supply	3.3 V only. No regulator, no level shifter	**Panel**	2.8 inch ST7789, 240 × 320, 40 MHz clock
Panel socket	FPC1, 12 way, 0.5 mm	**Backlight**	BSS138 (Q1), HIGH lights it
Holes	8, 2.54 mm pitch	**15-way**	15 ways, 1.0 mm FPC (U31)
Outline	48.6 × 79.8 mm, 1.6 mm board	**Mounting**	4 holes, Ø 4.8 mm. Printed V1.1